REMARKS

This amendment is a full and timely response to the Final Office Action mailed September 28, 2007. Reexamination and reconsideration are respectfully requested.

Applicant appreciates Examiner's acceptance of the amended drainings filed July 24, 2007.

Claims

Claims 1-12 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Applicant's Admitted Prior Art ("AAPA") in view of U.S. Pat. No. 5,410,651 to Sekizawa et al. ("Sekizawa"). This rejection is respectfully traversed.

Sekizawa teaches a distributed processing system in which a plurality of network-connected processing units performs computations and have the ability to transfer individual programs among each other utilizing network communications. (Sekizawa, Abstract). If an individual processor does not currently possess a required program in its local memory, it can request that a second processor that does have the required program in its memory transmit the program across the network. (Sekizawa, 4:18-5:3). Additionally, processors that are over- or under-utilized can request a transfer of programs between processors to more evenly distribute load. (Sekizawa, 5:38-7:4).

Applicant hereby renews and reiterates the arguments made in the Amendment filed July 24, 2007 in this application. Furthermore, the rejections set forth in the September 28, 2007 Office Action fail to establish a *prima facie* case of obviousness.

> To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. **Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations.**
>
> MPEP § 2143 (emphasis added)

As stated in the September 28 Action, the AAPA fails to teach that "the instruction code defined to be executed by the other processing units (second processor) is loaded separately from the instruction code defined to be executed by the one of the processing unit (first processor)." (September 28 Action, page 4). Specifically, with regard to this feature Applicant's claim 3 recites:

> a step of loading, by means of one of the central processing units of which the reset state is canceled, the common code and the instruction code defined to be executed by the one of the central processing units of which the reset state is canceled, **into the internal storage means** from the external storage means with the use of the direct memory access controller, by executing the boot code written in the boot storage means by the one of the central processing units of which the reset state is canceled, and loading, by means of one of other central processing units of which the reset state is canceled, the instruction code defined to be executed by the one of other central processing units of which the reset state is canceled, **into the internal storage means** from the external storage means with the use of the direct memory access controller by executing the boot code written in the boot storage means by the one of other central processing units of which the reset state is canceled;
>
> (emphasis added)

In other words, Applicant's claims recite loading the common code, the instruction code for the first processor, and the instruction code for the second processor separately into a single internal storage means.

Sekizawa may teach a plurality of processing units, each of which loads instruction codes into its own respective storage means separately. (Sekizawa, 2:56-3:12 and Fig. 5). However Sekizawa fails to teach or suggest a plurality of processors, each of which individually loads its respective instruction code into a single common internal storage means for execution. Sekizawa does not suggest *any* shared memory between the individual processing units at all. Separately loading executable codes for the plurality of processors, along with the common code, into a single shared internal storage means leads to implementation issues not at all addressed by Sekizawa. These issues are only resolved through considering the teachings of Applicant's specification. (*See e.g.* Applicant's specification, Fig. 2).

Thus, neither reference teaches or suggests loading instructions for execution two or more processors separately into a single storage means. The combined prior art references cited in the September 28 Action fail to teach or suggest all the claim limitations, and the September 28 Action fails to establish a *primae facie* case of obviousness for at least these reasons. Accordingly, Applicant respectfully requests reconsideration of the outstanding rejection of the pending claims.

In view of the above amendment, applicant believes the pending application is in condition for allowance.

Applicant believes no fee is due with this response. However, if a fee is due, please charge our Deposit Account No. 18-0013, under Order No. SON-2802 from which the undersigned is authorized to draw.

Dated: November 6, 2007

Respectfully submitted,

By_____
Ronald P. Kananen
Registration No.: 24,104
RADER, FISHMAN & GRAUER PLLC
Correspondence Customer Number: 23353
Attorney for Applicant